A Data-Driven Model of Pedestrian Movement

L. Casburn[1], M. Srinivasan[1], R.A. Metoyer[1], and M.J. Quinn[1]

We present a method for simulating individual pedestrian motion based on empirical data. Our model keeps track of the pedestrian's position and body configuration (pose) and uses motion capture data to produce plausible motion. While our ultimate goal is creating 3D animations of crowds, our initial efforts focus on 2D simulations. In this paper, we present a 2D model for an able-bodied male. Using our approach, we could also capture data and build models for a heterogeneous population, including children, the elderly, pedestrians in wheelchairs, and people on crutches. We demonstrate the realism of our model with a small-scale test case and a larger crowd evacuation simulation.

1. Introduction

The goal of our research is to demonstrate the feasibility of using pedestrian simulation for emergency planning, emergency response decisions, and training with respect to transportation facilities, sports arenas, and high-rise office buildings. In addition, we hypothesize that faithful computer models of pedestrian motion will be useful to architects designing new facilities. In particular, such a system will help them gain confidence that they have provided for the rapid egress of diverse populations in emergency situations. In order to be effective, training devices must be realistic enough that the users are able to "suspend disbelief." Realistic motion of individual pedestrians is a necessary condition for the suspension of disbelief. Several existing models produce pedestrian motion that is not realistic at the level of the individual pedestrian. For example, pedestrians may exhibit abnormally high velocities or unnatural direction changes. We argue that in order to build accurate models of how people move, we should observe real 3D human motion and build models from these observations. We hypothesize that the more realistic the motion is, the more accurate the results of the simulation will be.

In this paper, we describe a new model of 2D pedestrian motion. We begin by collecting 3D motion capture data from a human subject. We maintain a 3D kinematic model of each pedestrian throughout the simulation. Every "pose" (configuration of joints) adopted by a pedestrian is consistent with the motion capture data we have collected. Likewise, every pedestrian movement generated by the model is consistent with the motion capture data. At any moment in the simulation, a pedestrian has the capability to reach a variety of different spatial points at some future time. The behavior of the pedestrian is reflected by its particular choice of these spatial points. Hence we call our approach the capability-behavior model of pedestrian motion.

[1] School of Electrical Engineering and Computer Science,
Oregon State University, Corvallis, Oregon, USA

2. Related Work

Many other microsimulation models of pedestrian motion exist. For an excellent survey of these models, see Helbing et al. [3]. One way to categorize microsimulation models is to divide them into cellular automata models, agent-based (or AI-based) models, and behavioral force models.

The principal advantage of cellular automata models is that they execute very fast on computers. However, because cellular automata models divide space into cells, motion is discrete, undermining our goal of creating training simulators that will encourage participants to suspend disbelief.

Behavioral force models accurately predict a variety of phenomena that occur when crowd densities are high. However, they have a couple of issues which make them problematic for us. First, pedestrians inside dense clusters exhibit an unnatural "jitter." Also, our experience with this model has been that there is no single set of parameters that works in every instance. Instead, the parameters must be carefully tuned in order to fit the scenario being modeled. Finally, it is unclear how this force-based approach would be able to model pedestrians with restricted motion abilities. For example, a person in a wheel chair cannot instantaneously shift sideways.

Agent-based models allow each pedestrian to have a unique behavior. Hence modeling a heterogeneous population is much easier with an agent-based model. Agent-based models have shown good success when pedestrian densities are low to moderate. They have the liability that the computational demands for simulating an agent can be high. We categorize our capability-behavior model as an agent-based model.

Human motion generation has been a key problem in computer graphics for a number of years. Most recently, research in motion graph techniques has led to algorithms for generating natural human motion that satisfies some user constraints, such as trajectory tracking [1, 6, 7, 8]. Another important area of computer animation and simulation is level of detail generation. Creating multiple levels of detail allows for efficiency optimizations when accuracy can be traded off for speed. Most closely related to our work is that of Brogan et al., in which they have built capability models that describe what motions a physical simulation is capable of achieving [2]. In our case, we want to build capability models based solely on motion capture data, because we believe such models will result in more realistic motion patterns than current approaches.

3. Motion Capture

Our goal in this paper is to create a 2D motion model that encodes the capabilities of a walking human. This model should take into account not only the physical characteristics of the person (size, weight, disabilities, etc), but also the current state of the person when making movement decisions. For example, is the person standing still or in the middle of a walking stride? To do so, we will build our model from real human motion data recorded with motion capture technology.

Motion capture technology has been used in biomechanics and entertainment for many years as a means of recording human motion. There are several different motion capture technologies. Magnetic and optical systems are the two most commonly used approaches in computer graphics and entertainment.

For this work, we used a Vicon 612 optical motion capture system [10]. An optical motion capture system consists of three main components: markers, sensors, and a workstation.

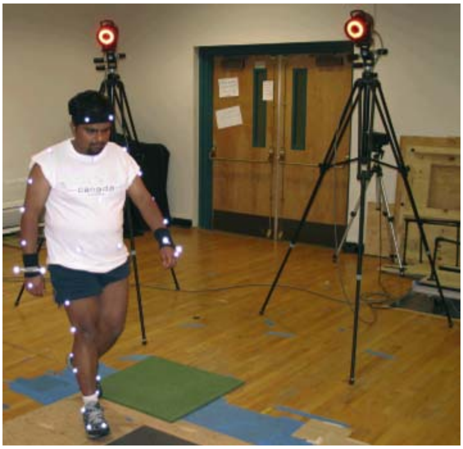

Figure 1: Highly reflective optical markers placed on a motion capture subject.

The markers consist of highly reflective spheres, 5 mm in diameter, that are placed at strategic locations on the subject to be recorded (Figure 1). The sensors consist of several high speed infrared cameras. Each camera is placed on the perimeter of an area around the subject of interest and aimed toward the center of the activity. The intersection of the camera view frustums forms the capture region. As the subject moves within the capture region, the camera images are recorded on the workstation.

The Vicon system processes the recorded data to produce 3D locations for each marker. As long as a marker is visible in the images of any two cameras, the 3D location of that marker can be computed. This 3D location information for each marker is called the raw marker data.

To generate animated motion, or in our case, models of human motion, we convert the data to a more usable form. Using the Vicon software suite, the raw marker positions are converted to joint angles according to a kinematic model of the subject.

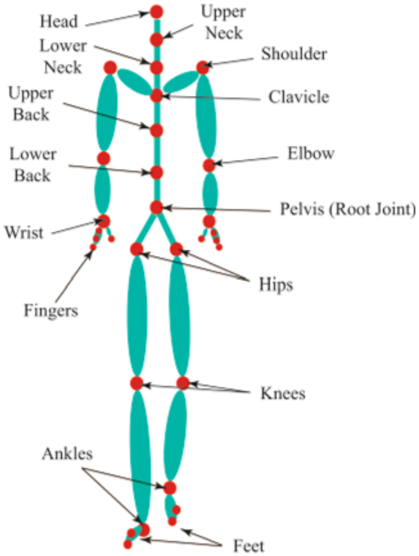

Figure 2: Our kinematic human model consists of 30 joints, each with three degrees of freedom.

For all of our experiments, we used the Vicon 612 3D optical motion capture system with 6 cameras and captured the data at a rate of 60Hz. The raw marker information was converted to joint angles for a kinematic subject model consisting of 30 joints (Figure 2).

4. Motion Graphs and Mobility Maps

Once the joint angles are computed for each frame of motion, we further process the data to build a data structure known as a motion graph. Motion graphs have recently been used in the computer animation field to generate synthetic human motion by drawing upon a library of previously recorded human motion. In the following sections we will give a brief overview of motion graphs. For a more detailed description, see [1, 6, 7, 8].

Using our Vicon optical motion capture system, we capture the 3D motion of subjects performing various tasks relevant to pedestrians: walking straight at various rates, turning at various radii, coming to a stop, starting from a stop, and turning in place. We then build a graph that encodes natural transitions between these sequences of motion.

The recorded motion segments are first combined into a single large collection of poses, where a pose is defined as the set of angles for each degree of freedom, as well as the root body position and orientation for a single frame of motion. Each frame, therefore, consists of 96 floating point values, representing the three Euler angle values for each of the 30 joints and the 6 degrees of freedom of the root body. In our experiments, the root body is centered at the pelvis (Figure 2).

We define a distance metric that measures the similarity of two poses, and we use this metric to develop a pose transition cost. The pose transition cost function determines the cost to move from one pose to another. If a pose is likely to follow another pose in natural motion sequences, the cost is low. We build a fully connected graph, where each node in the graph is a pose from the original motion data, and each arc in the graph is the cost to transition between the poses connected by the arc. The graph is then pruned to remove all transitions that are above a threshold value (i.e., the unrealistic transitions). We call the resulting graph a motion graph.

The motion graph encodes natural transitions, such as transitions from poses that come from walking in a straight line to poses that come from turning. Current motion graph techniques typically search this graph for an optimal path that adheres to some constraint, such as following a 2D trajectory or reaching a desired point on the 2D floor plane. The result is a sequence of poses that meets the constraints while maintaining smooth transitions. These search-based approaches result in compelling motion for the given trajectory but are fairly expensive. Since we would like to generate motion at interactive rates, search based approaches are not sufficient.

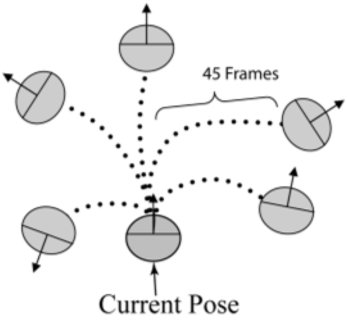

Figure 3: Illustration of movement options from a single pose in the mobility map. Each pose reachable from the current pose in 45 frames is called a movement option and is stored along with the sequence to reach it, the final position, and the final orientation with respect to the current pose.

For this reason, we further process the motion graph to produce a data structure that sacrifices memory space for speed. For each pose in the graph, we compute the shortest path between it and every other pose in the path. We call this the "all pairs shortest path" (APSP) graph. For each pose in the APSP graph, we collect all nodes that can be reached in some small time horizon (1.5 Seconds = 45 frames). For each reachable node, we store the final pose as well as the sequence of intermediate poses that lead to that pose and the relative change in position and orientations (Figure 3).

This last step results in what we call a mobility map. The mobility map can be viewed as an oracle that the simulated pedestrian can consult to understand its capabilities given its current state. The mobility map can tell the pedestrian all locations it can reach, how it will be oriented, and the sequences of poses to get it there. For a more detailed description of mobility maps, see [9].

Using mobility maps, one can produce compelling 3D motion at interactive rates for up to 500 characters. However, the pedestrians sometimes suffer from a wandering behavior. Wandering occurs when a pedestrian cannot achieve a spatial request because the database of motions simply does not contain enough of the right kind of data. In other words, there is no smooth sequence of poses that quickly brings the pedestrian from its current location and pose to the desired location. This smoothness constraint is very important when generating 3D motion, because viewers are very good at noticing small discontinuities in synthetic human motion.

In this paper, we modify the mobility map approach by taking advantage of the fact that we are not creating 3D animations. Rather, we are concerned with developing 2D human motion models. This goal allows us to relax the 3D motion smoothness constraint. In doing so, we can increase the controllability of the pedestrian by clustering similar poses and building a more coarse-grained mobility map from the clustered poses. The result is a clustered mobility map, where each pose is replaced by a cluster of poses with a larger number of movement options. Since a single mobility map can be shared by all ‘similar’ pedestrians (e.g., all able-bodied male pedestrians, ages 30-35) in the scene, space usage scales linearly with the number of pedestrian types, and not with the number of pedestrians.

5. Clustered Mobility Maps

A simulated pedestrian is capable of reaching a number of spatial locations, which we call mobility points, within a time step of 1.5 seconds. We set the time step to 1.5 seconds because this is long enough to provide the pedestrian with a reasonable number of mobility points, but short enough to keep the memory requirements reasonable. The nature of the motion is determined by the current body posture or pose. For instance, if a pedestrian is standing in a pose with two feet on the floor it is likely that it will continue to stand or begin to walk at a low velocity. If the pedestrian is in a pose with one foot on the floor, the pedestrian may be in a position to move more quickly because it is in the middle of a walking stride.

Similar poses should also exhibit similar dynamics. Motion sequences starting from a pose with a left heel down and right toe pushing on the floor, for example, should share the same characteristics as motion sequences generated from a different pose that also has its left heel down and right toe on the floor. Therefore, we group similar poses together into a single cluster. In order to identify groupings of similar poses, we use a subtractive clustering algorithm described by Kim et al.[5]. For determining clusters for basic able-bodied locomotion, the upper body degrees of freedom contribute very little information and therefore are ignored. We consider only the degrees of freedom from the waist down.

Once the poses have been clustered, we can combine the individual original mobility maps for each pose in a cluster into a single mobility map for the cluster. The cluster shares the movement options from all the poses within it, thus giving a simulated pedestrian more mobility and greater ability to react to movement requests. In the following section, we will describe how we use this new clustered mobility map to drive the 2D simulation.

6. Capability-Behavior Model

At run-time, we use the clustered mobility map to animate the motion of each 2D pedestrian. We assume that the pedestrian simulator makes target requests for each pedestrian. We define a target to be a position a pedestrian wants to reach, such as the location of the nearest exit. Pedestrians exit the building by moving toward the targets. At any point in time, a pedestrian is in a particular pose of a cluster and will have movement options dictated by the mobility map of that cluster.

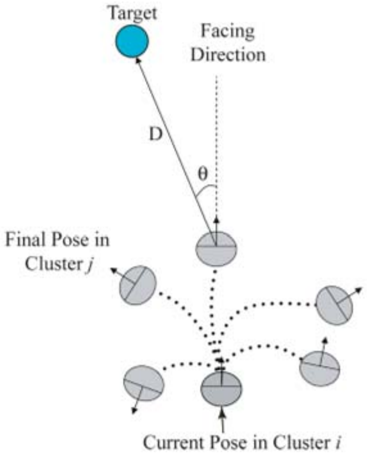

Figure 4: The new clustered mobility map is combined with a cost function to rank each option. The cost of a movement option depends on the distance from the target and the deviation between the facing direction and the target direction.

Given a target request, the movement options are first ranked based on a cost function:

$$C_i = \omega_D D + \omega_\theta \theta - \omega_S S + \sum \omega_P P \tag{1}$$

The cost function is similar to that presented by Srinivasan et al. [9], where D measures the Euclidian distance from a mobility point to the target and θ is the deviation angle (Figure 4). The deviation angle is the angle between the facing direction a pedestrian has upon taking a movement option and the vector from the mobility point to the target. We modify the function to include a term, S, which encourages faster movement, and another term, P, which incorporates interference from neighboring pedestrians. S represents the displacement from the current position when taking a movement option. Options that move the pedestrian farther in 1.5 seconds will have a lower cost.

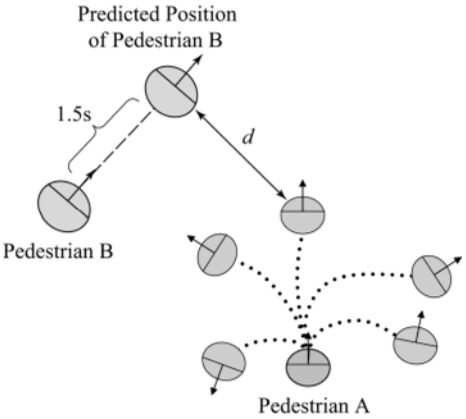

Figure 5: Illustration of pedestrian interference factors. To determine the cost due to nearby pedestrians, we consider the distance from the mobility point to the predicted position of the nearby pedestrian.

The last term in the cost function measures the influence of neighboring pedestrians (Figure 5). We represent this influence with a function similar to that used by Helbing and Molnar in the social forces model [4]. The closer the mobility point is to another pedestrian the higher the cost. If, however, the pedestrian is moving at a slower velocity, it will be more comfortable selecting a move that is close to another pedestrian. We represent the cost with a decreasing exponential function:

$$P = a * e^{-d/(b*v)} \tag{2}$$

where d is the distance between the pedestrian at the mobility point and the predicted position of the other pedestrian. To compute the predicted position of other pedestrians, we assume constant velocity over the time step. The velocity of the pedestrian at the mobility point is represented by v, while a and b are constants with values 10.5 and 0.09, respectively. These values were determined through trial-and-error and used throughout all of our experiments. The total pedestrian influence for that movement option is the summation of the influence from each of the neighboring pedestrians.

Variables ω_d, ω_θ, ω_s, and ω_P are weighting terms for the distance, deviation angle, speed, and pedestrian interference terms, respectively. Pedestrians that are behind the mobility point are not considered to be interfering, therefore $\omega_P = 0.0$, otherwise $\omega_P = 1$. We use a grid spatial subdivision scheme to determine nearby pedestrians efficiently.

We rank the movement options according to their cost and store them in a priority queue. The first option in the priority queue has the least cost. Our algorithm iteratively extracts the options from the queue until one is found that does not intersect with walls or nearby obstacles. We implemented a quadtree subdivision scheme to efficiently determine nearby walls and obstacles for intersection tests. This movement option is chosen as the desired option; it is used to animate the pedestrian motion for the next 1.5 seconds (45 frames). If the algorithm cannot find a suitable option, we allow our pedestrian to linearly interpolate its position and orientation, for four frames, toward its goal. This allows the pedestrian to move into a position or face away from the obstacle so that at least one movement option can become feasible. In our experiments, the simulated pedestrians resort to the interpolation option only 2% of the time.

In summary, the cost function leads the simulation to prefer movement options that quickly move the pedestrian closer to the target, orient the pedestrian toward the target, and have the least amount of interference with other pedestrians.

7. Results

These experiments demonstrate that the capability-behavior model can create realistic-looking crowd motion. Even when moving in a straight line, simulated pedestrians have a bit of side-to-side motion characteristic of actual pedestrian movement. Pedestrians spread out and avoid collisions when moving through an area where there is plenty of room. At bottlenecks, they bunch up without intersecting. Simulated pedestrians within crowds stand quietly, without any of the "jitter" characteristic of the social forces model (Figure 6). The capability-behavior model parameters are identical for all the examples we have presented; no "tuning" of parameters to fit the particular situation was required.

Figure 7 illustrates the use of the capability-behavior model to simulate the orderly evacuation of a nearly-full auditorium, along with several adjacent offices. This example demonstrates the ability of our current simulator to move pedestrians in a more complicated setting. Two simplifying assumptions reduce the realism of this simulation. First, every pedestrian begins moving at the same time. Second, every pedestrian has the goal of leaving the building through the nearest exit. In this simulation the total egress time was 316 seconds.

To date, we have only collected motion capture data for a single person. About 20 minutes of samples resulted in a total of 33,404 different poses. The clustering algorithm collected these poses into 670 clusters. In the experiments reported in this paper, the movements of every simulated pedestrian are drawn from the same set of mobility maps. Even with this limitation, we believe the resulting crowd movements are plausible. Collecting additional motion capture data will enable us to create a greater diversity of simulated pedestrians, adding to the overall realism of the results produced by the system.

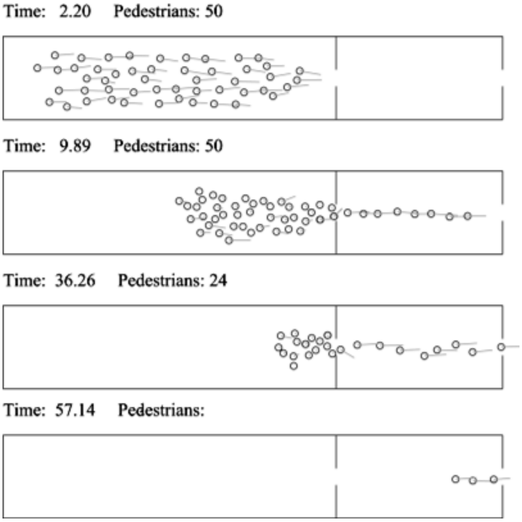
Time: 2.20 Pedestrians: 50
Time: 9.89 Pedestrians: 50
Time: 36.26 Pedestrians: 24
Time: 57.14 Pedestrians:

Figure 6: Snapshots of the doorway experiment at simulated times (sec) of 2.2, 9.89, 36.26, and 57.14. The pedestrians must approach the door, move through the door, and proceed to the second door.

8. Future Work

There are several avenues of research remaining for future work. First, we plan to capture motion of pedestrians with varying capabilities to begin building a diverse database of 2D motion models. We hope to capture individuals with crutches, canes, in wheelchairs, and of varying ages (children and elderly), for both males and females.

Ultimately, we intend to create models that can be used in real-time simulators. We are currently investigating statistical techniques for improving the efficiency of our approach in terms of memory requirements and CPU time. We hope to eliminate the need to store and search all movement options and replace them with movement option distributions that encode the capabilities of a pedestrian from a given pose with statistical measures.

Finally, we are investigating the addition of higher level behaviors, such as route choices, grouping, and collision negotiation and avoidance, into our overall pedestrian simulation system. These behaviors will be designed to sit on top of the presented motion model.

The behaviors will be included in the cost function and will therefore lead to the choice of movement options appropriate for the desired behavior.

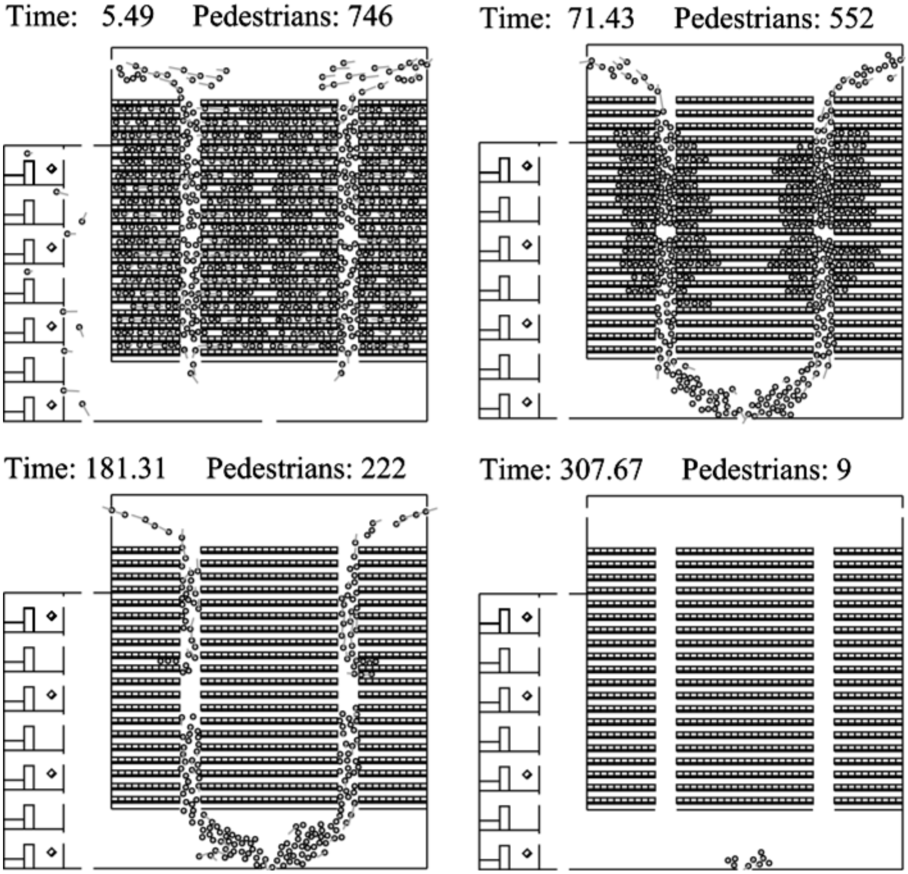
Time: 5.49 Pedestrians: 746
Time: 71.43 Pedestrians: 552
Time: 181.31 Pedestrians: 222
Time: 307.67 Pedestrians: 9

Figure 7: Four snapshots of a simulation of 750 persons evacuating an auditorium and adjacent offices.

9. Acknowledgments

The authors would like to thank Dr. Mike Pavol, Director of the Biomechanics Laboratory at Oregon State University, and Hong Zhu for their invaluable help with motion data collection and processing. This work has been supported in part by NSF CCR-0237706 and CNS-0423733. «

References

1. O. Arikan and D. A. Forsyth: *Interactive Motion Generation from Examples*, ACM Transactions on Graphics 21, pp. 483–490 (2002).
2. D. Brogan and J. Hodgins: *Simulation Level of Detail for Multiagent Control*, In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems, ACM Press, pp. 199–206 (2002).
3. D. Helbing, I.J. Farkas, P. Molnar, and T, Vicsek: *Simulation of Pedestrian Crowds in Normal and Evacuation Situations*, In: M. Schreckenberg and S.D. Sharma (Eds.), Proceedings of the International Conference on Pedestrian and Evacuation Dynamics, Springer, Berlin, pp.21-58 (2002).
4. D. Helbing and P. Molnar: *Social Force Model for Pedestrian Dynamics*, Physical Review, 51(5), pp. 4282-4286 (1995).
5. T. Kim, S.I. Park, and S.Y. Shin: *Rhythmic-Motion Synthesis Based on Motion-Beat Analysis*, ACM Transactions on Graphics, 22, pp. 392-401 (2003).
6. L. Kovar, M. Gleicher, and F. Pighin: *Motion Graphs*, ACM Transactions on Graphics, 21, pp. 473–482 (2002).
7. J. Lee, J. Chai, P.S.A. Reitsma, J.K. Hodgins, and N.S. Pollard: *Interactive Control of Avatars Animated with Human Motion Data*, ACM Transactions on Graphics, 21, pp. 491–500 (2002).
8. R. Metoyer: *Building Behaviors with Examples*, Ph.D. Thesis (2002).
9. M. Srinivasan, R. Metoyer, and E. Mortensen: *Controllable Real-Time Locomotion using Mobility Maps*, In: Proceedings of Graphics Interface, Victoria, B.C, pp. 51-59 (2005).
10. Vicon, *Vicon 612 optical motion capture system*, http://www.vicon.com (2004).